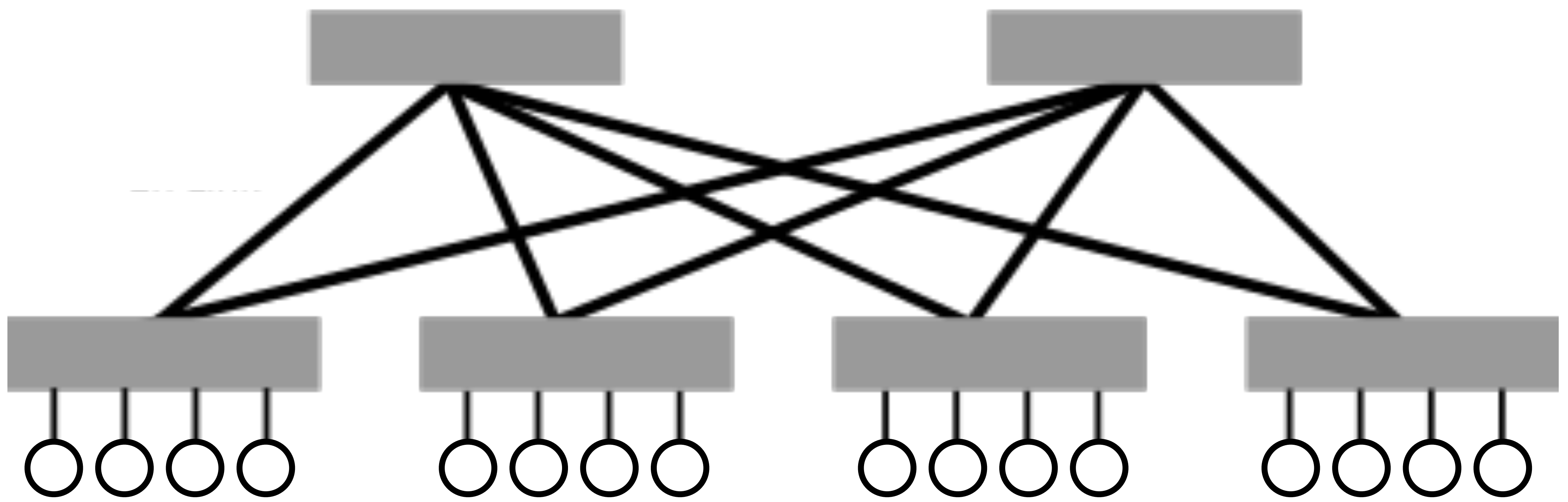
Graph optimization algorithm for low-latency indirect network

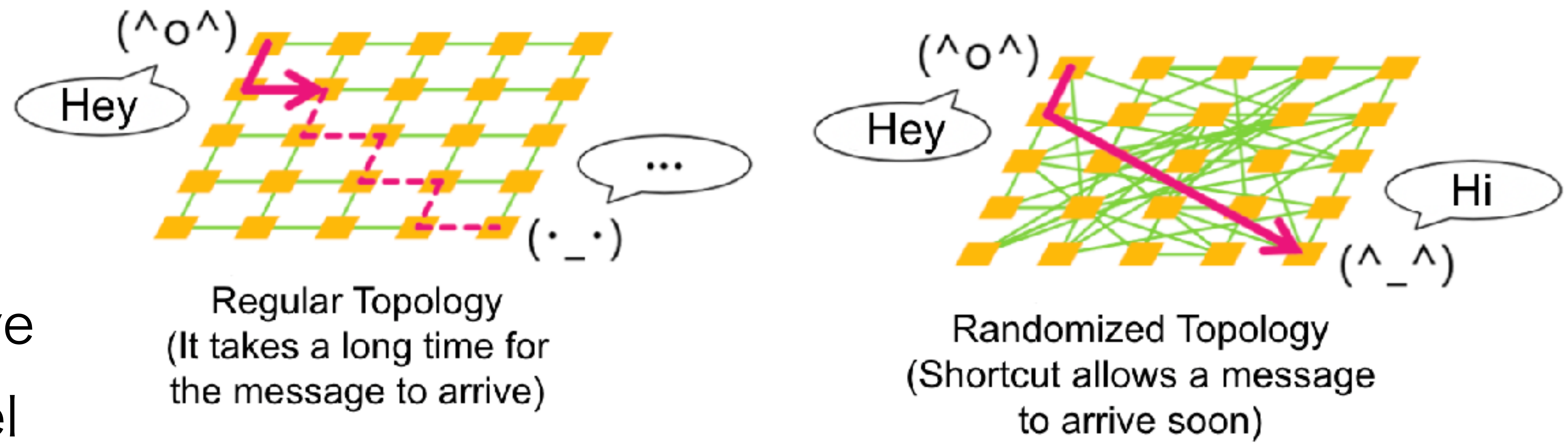
Masahiro Nakao[†], Masaki Tsukamoto[‡], Yoshiko Hanada[‡], Keiji Yamamoto[†]

† RIKEN R-CCS
‡ Kansai University

Background

- Network topology of a parallel computer system has a strong impact on the performance of parallel applications
- Regular topology : Fat-tree, k-ary n-cube, Dragonfly, Slim Fly, etc.
- **Random topology**
  - M. Koibuchi et al. A case for random shortcut topologies for HPC interconnects, 2012
  - It has been reported to improve performance of various parallel applications

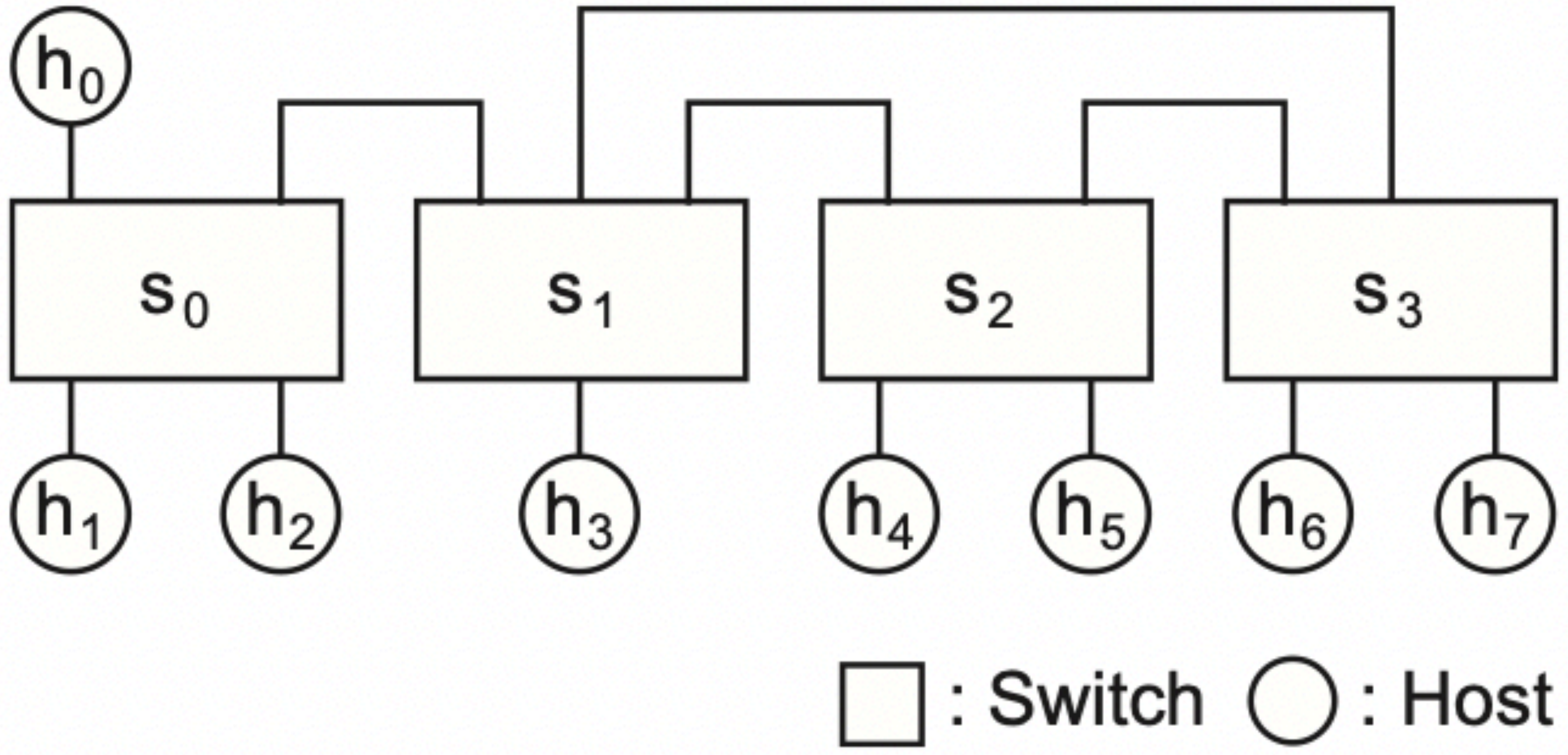




https://www.nii.ac.jp/userdata/shimin/documents/H27/150820_2ndlec.pdf

Order/Radix Problem (ORP)

- Research on indirect network of a parallel computer system based on graph theory
- Graph consists of vertices and edges
- By regarding the host and switch as vertices and the cable as edges, an indirect network can be represented in a graph
- **ORP is the problem of finding a graph with the minimum average host-to-host distance (h-ASPL) that satisfies a given “#hosts" and "radix"**
- #switches is arbitrary
- A host can only be adjacent to a switch, a switch can be adjacent to a host or switch



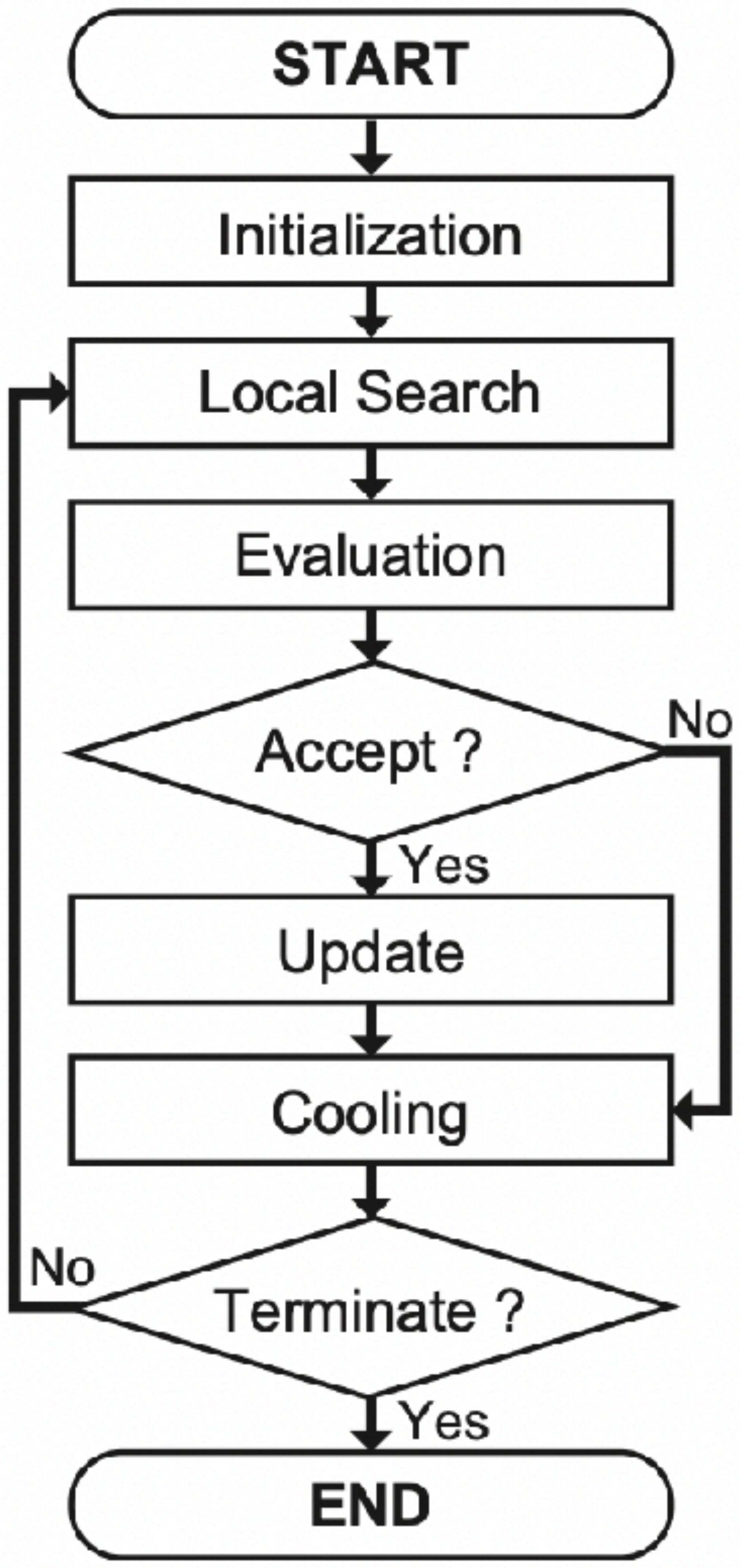
#hosts : 8, radix : 4, #switches4

h	0	1	2	3	4	5	6	7
0		2	2	3	4	4	4	4
1			2	3	4	4	4	4
2				3	4	4	4	4
3					3	3	3	3
4						2	3	3
5							3	3
6								2
7								

Distance matrix

h-ASPL = 91/28 = 3.25

Simulated Annealing

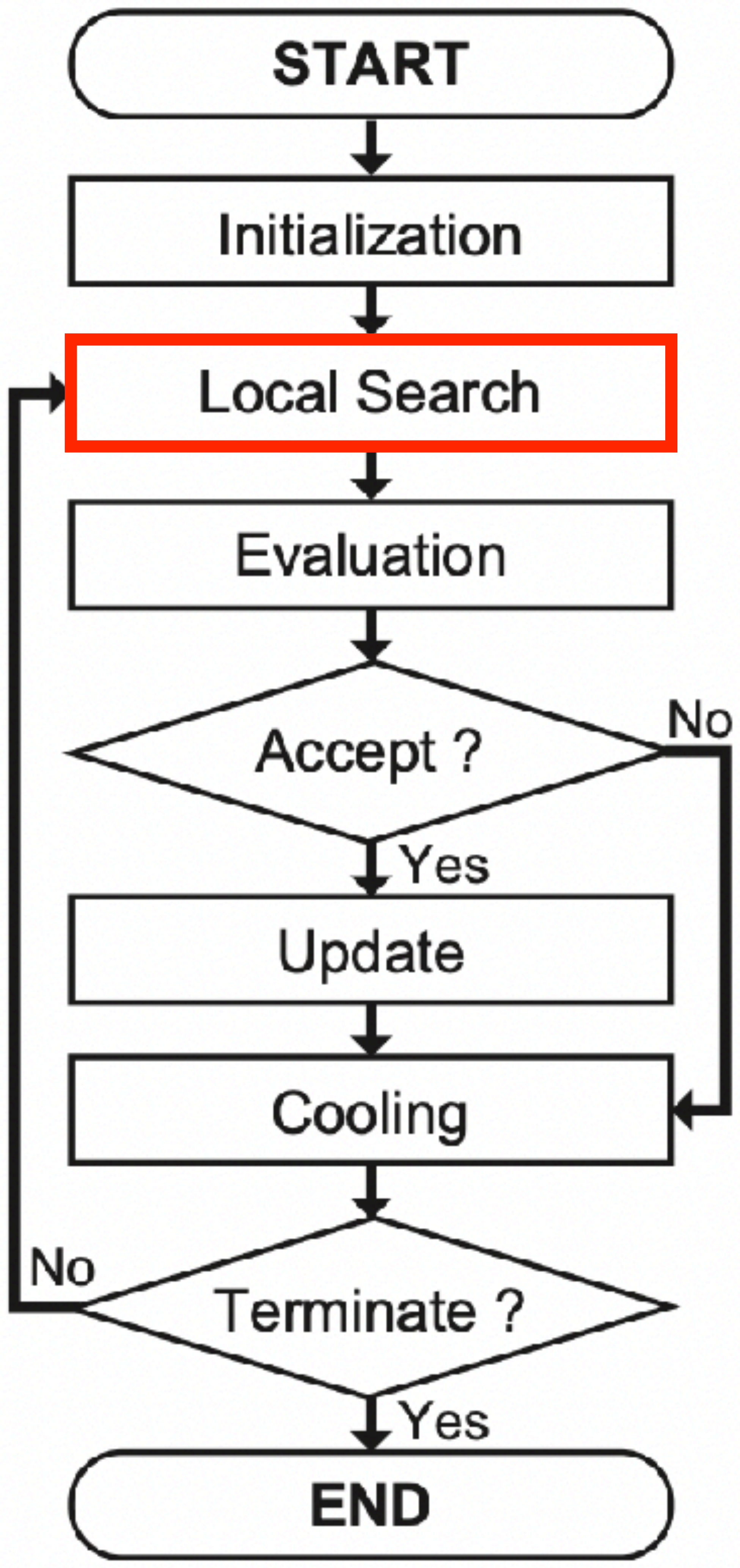


- Initialization : Create a random graph and set temperature
- Local Search : Change the graph
- Evaluation : Calculate h-ASPL
- Accept : Judge whether to accept the new graph or not
  - h-ASPL becomes smaller : must accept
  - h-ASPL becomes larger : According to the following probabilities

$$Probability = \begin{cases} 1 & \text{if } \Delta E' < 0 \\ \exp(-\frac{\Delta E'}{T}) & otherwise \end{cases}$$

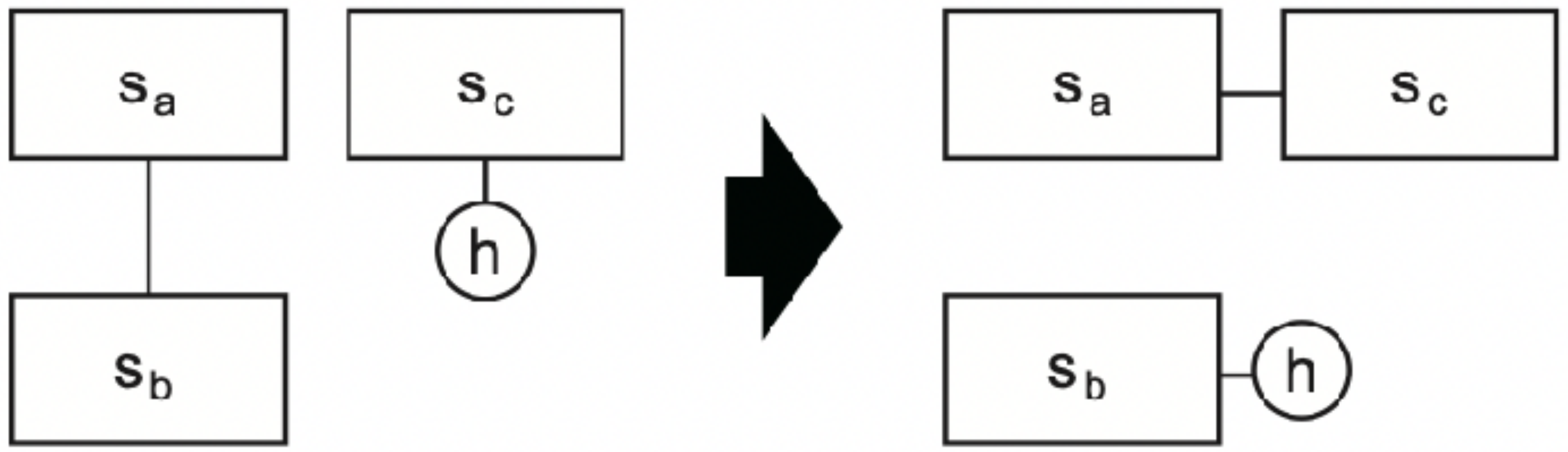
- Update : Replace with the new graph
- Cooling : Lower temperature
- Terminate : Finish after repeating the specified number of times

Local Search

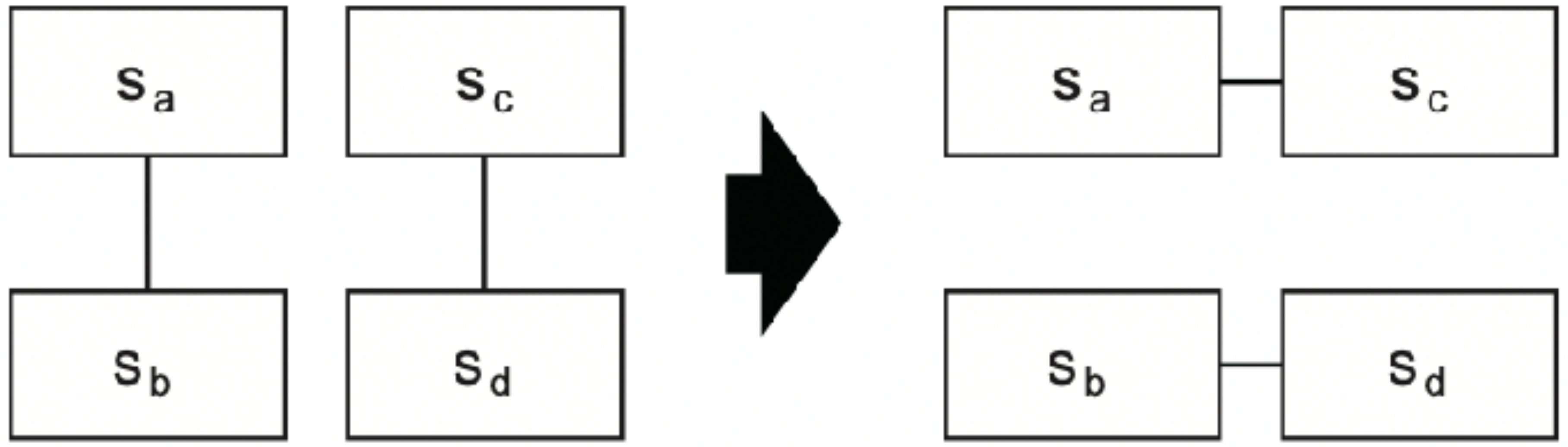


- **SWING**
  - Move a host to a different switch
  - Replace one edge

s : Switch, h: Host



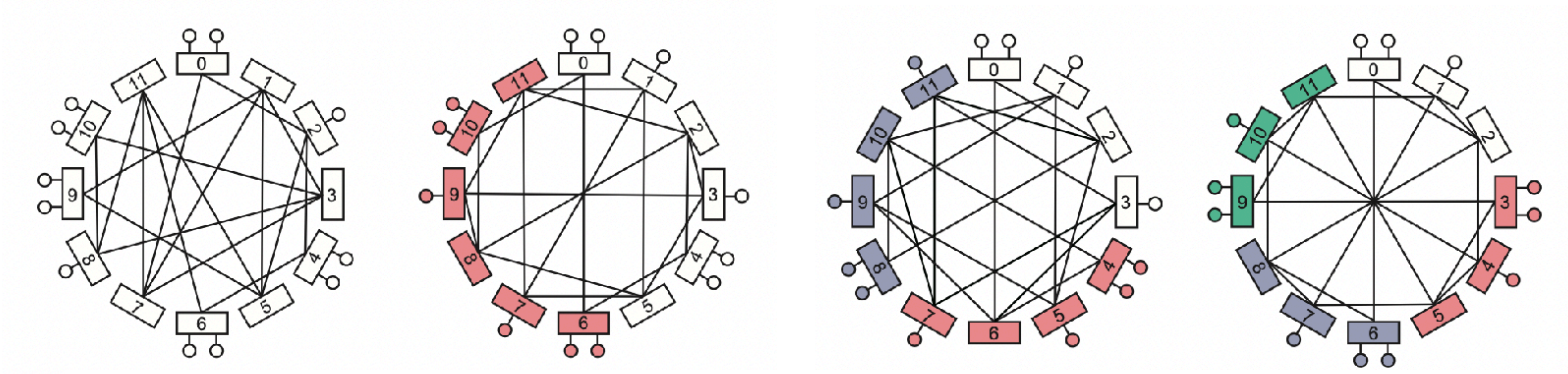
- **SWAP**
  - Replace two edges



The number of vertices adjacent to the switch (the number of hosts and switches) does not change

Simulated Annealing with symmetry

- Examples of graphs with symmetry (#hosts=12, radix=4, #switches=12)



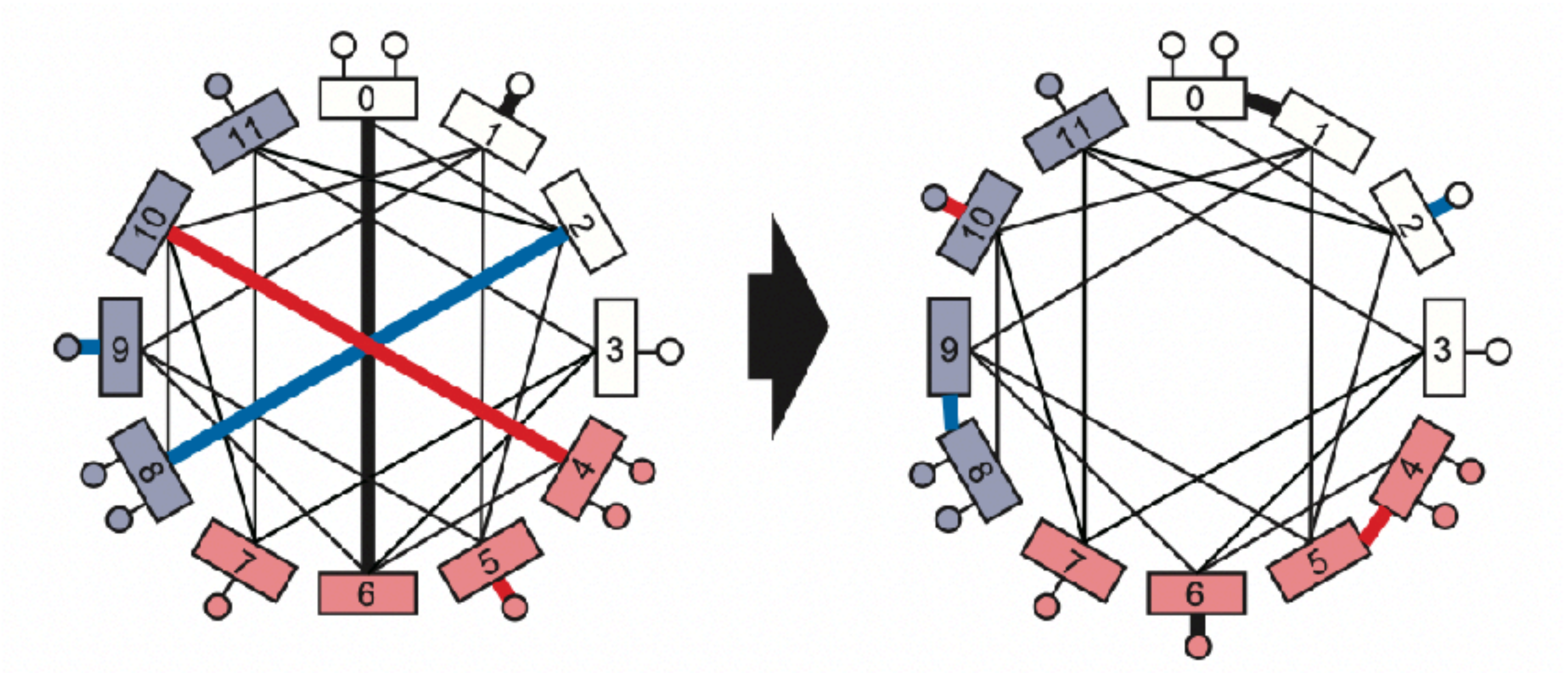
g=1 **g=2** **g=3** **g=4**

- When rotated 360/**g** degrees, the relationship between the vertices (host and switch) and the edges becomes the same
- The variable **g** is the common divisor of #hosts and #switches
- No symmetry when **g** = 1

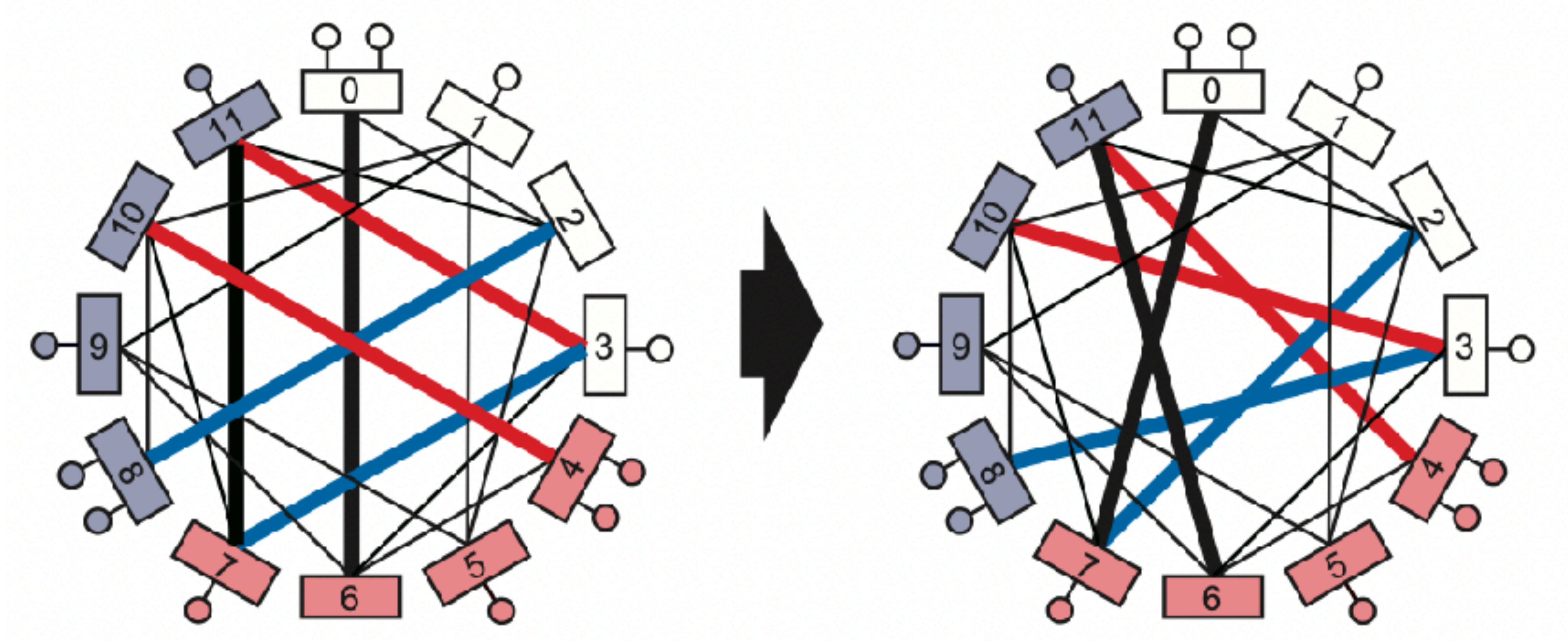
Local Search with symmetry

- Examples of graphs with symmetry (#hosts=12, radix=4, #switches=12, g=3)
- The initial solution is a graph with symmetry
- Perform SWING and SWAP on all vertices and edges that have a symmetric relationship

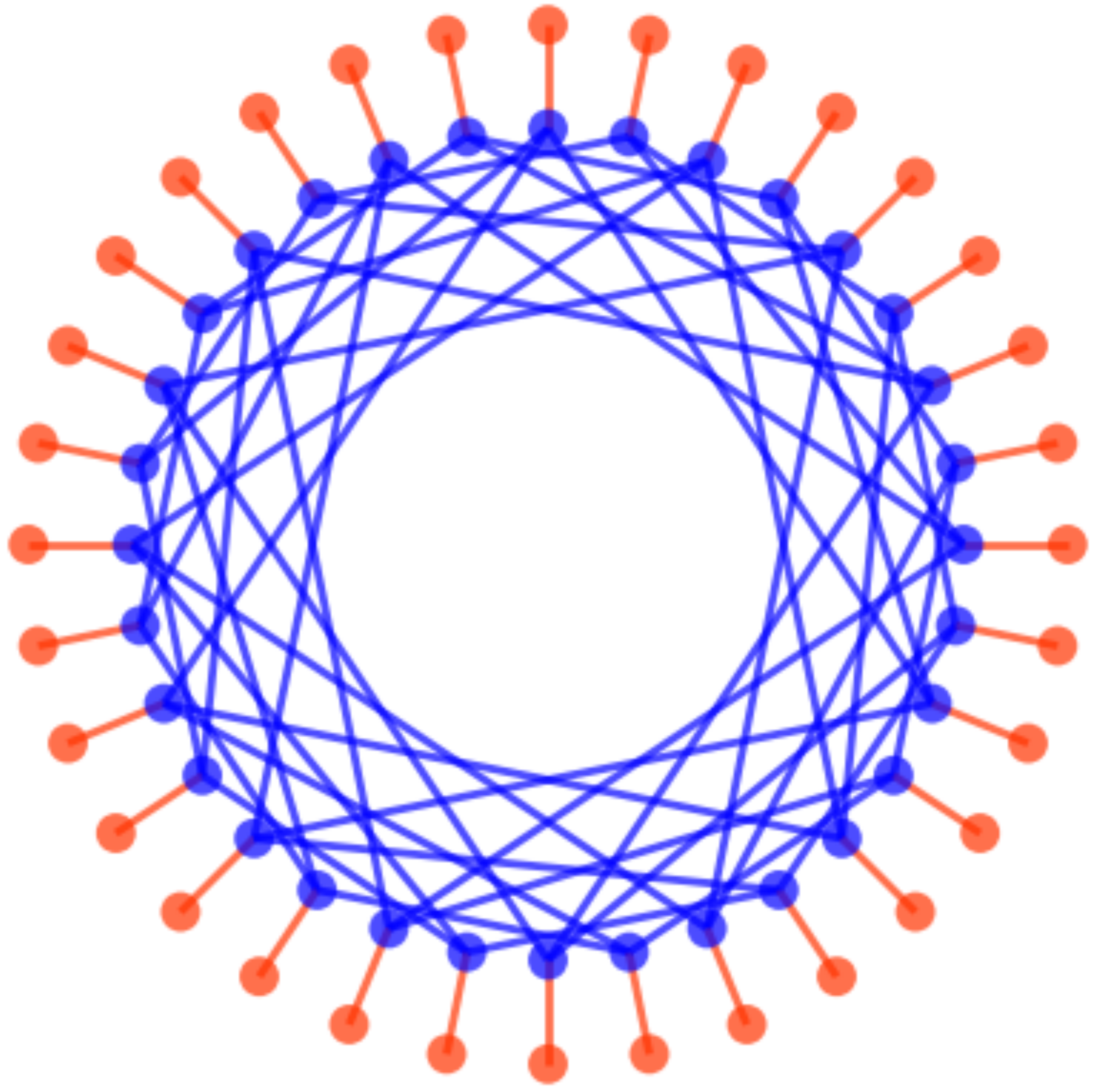
- **SWING**



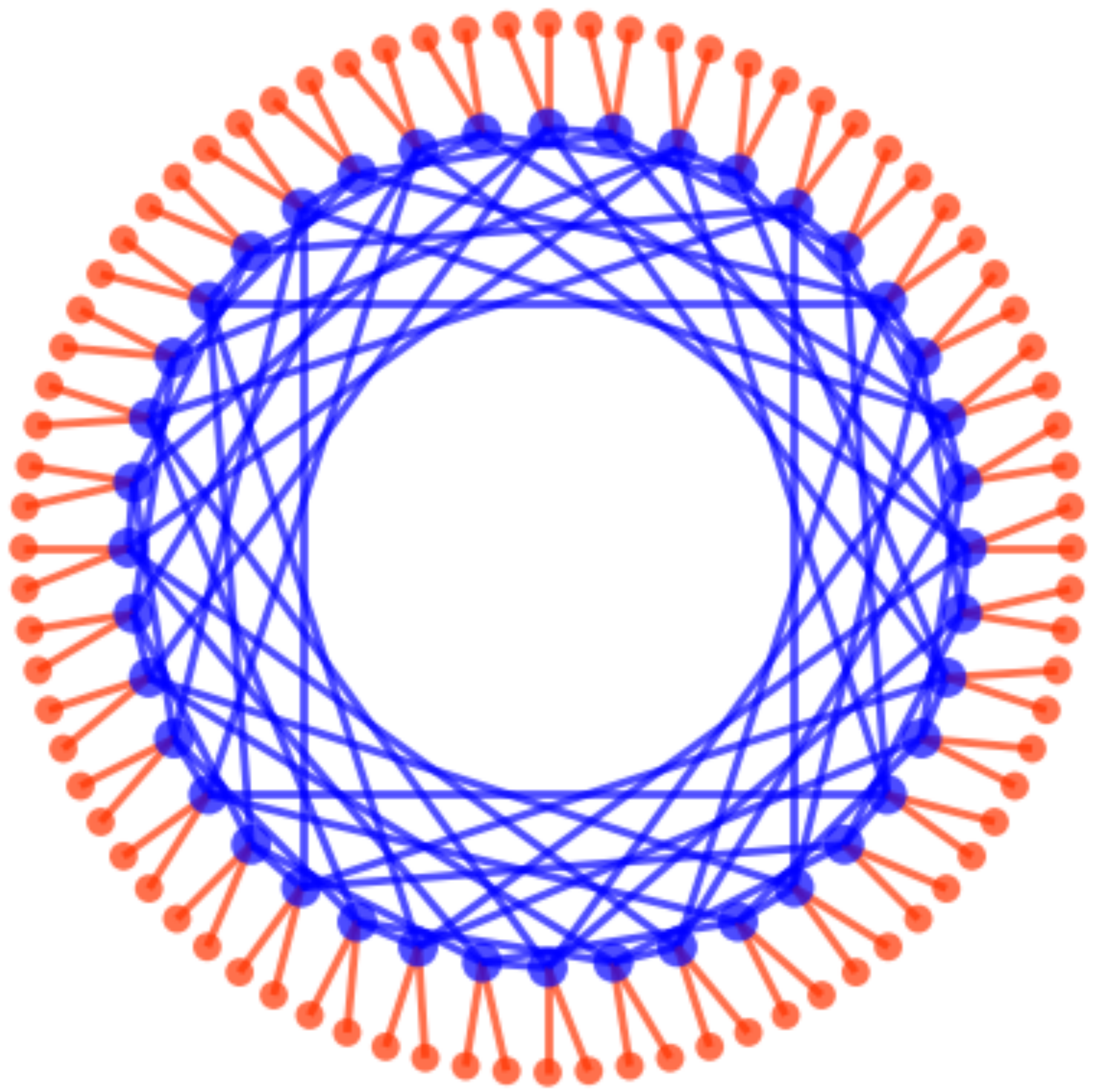
- **SWAP**



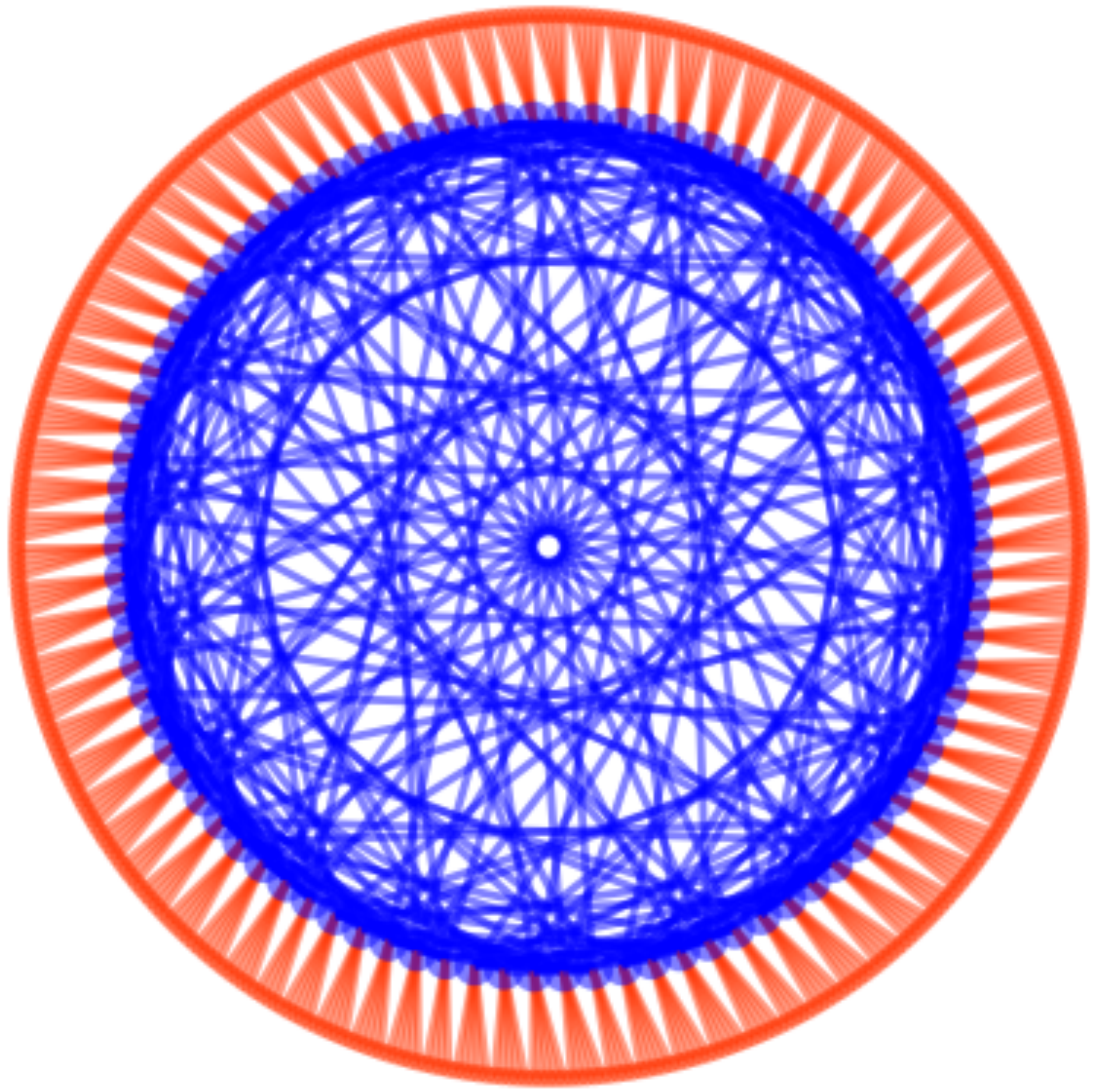
Example results (1/3)



hosts = 32
radix = 4
switches = 32
g = 16



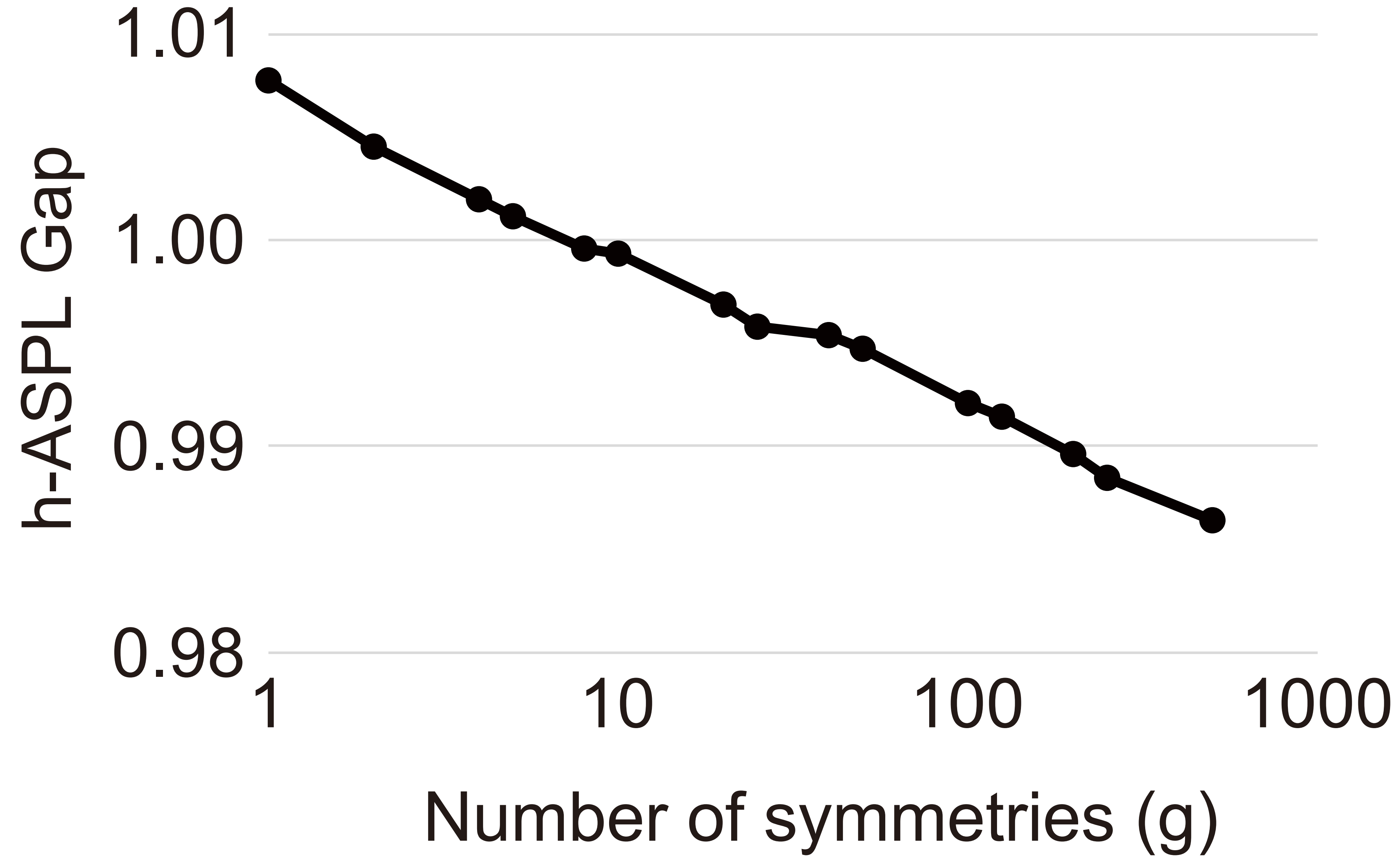
hosts = 80
radix = 6
switches = 40
g = 20



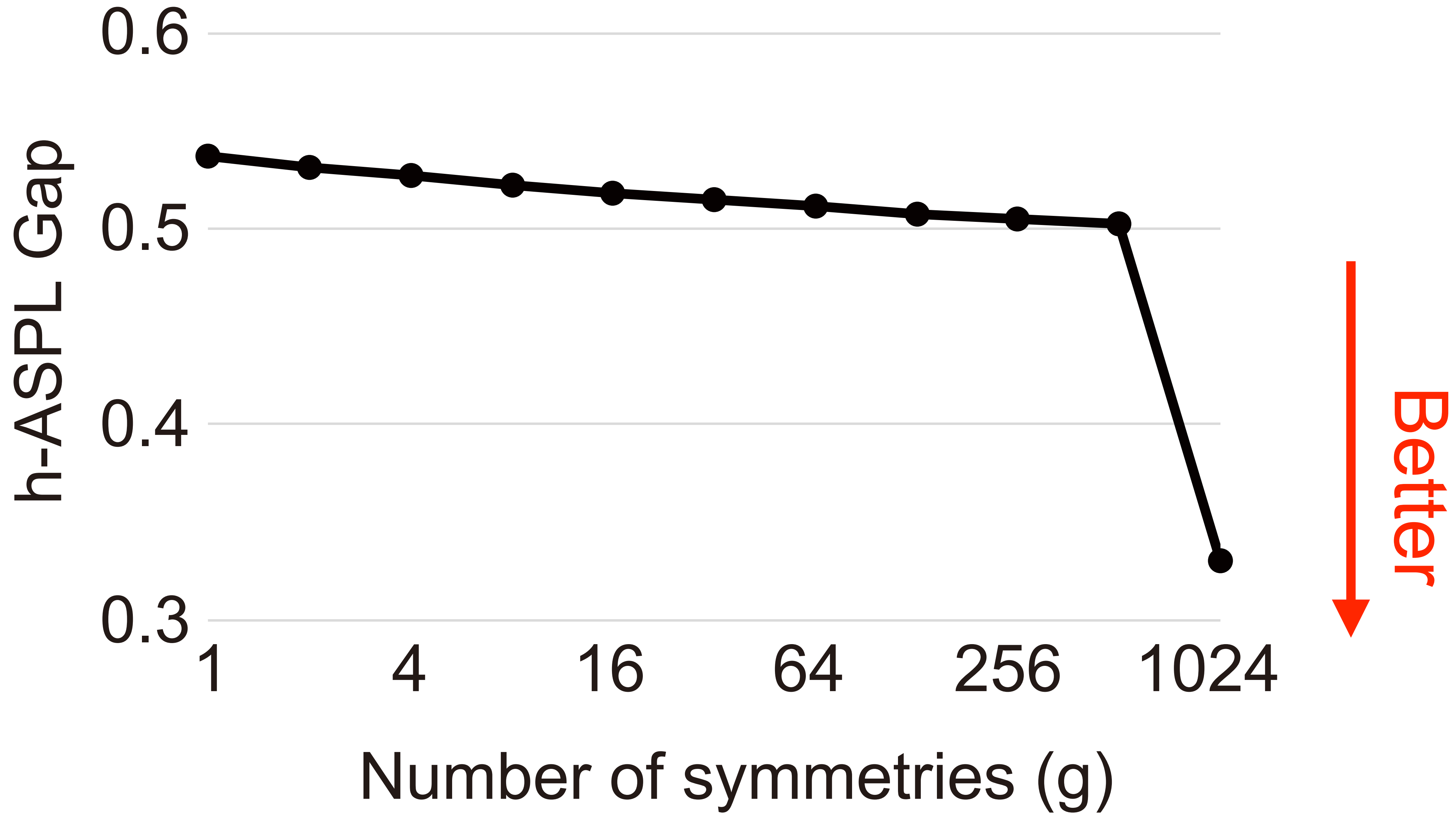
hosts = 432
radix = 12
switches = 90
g = 18

Example results (2/3)

#hosts = 10000, radix = 10
#switches = 5000



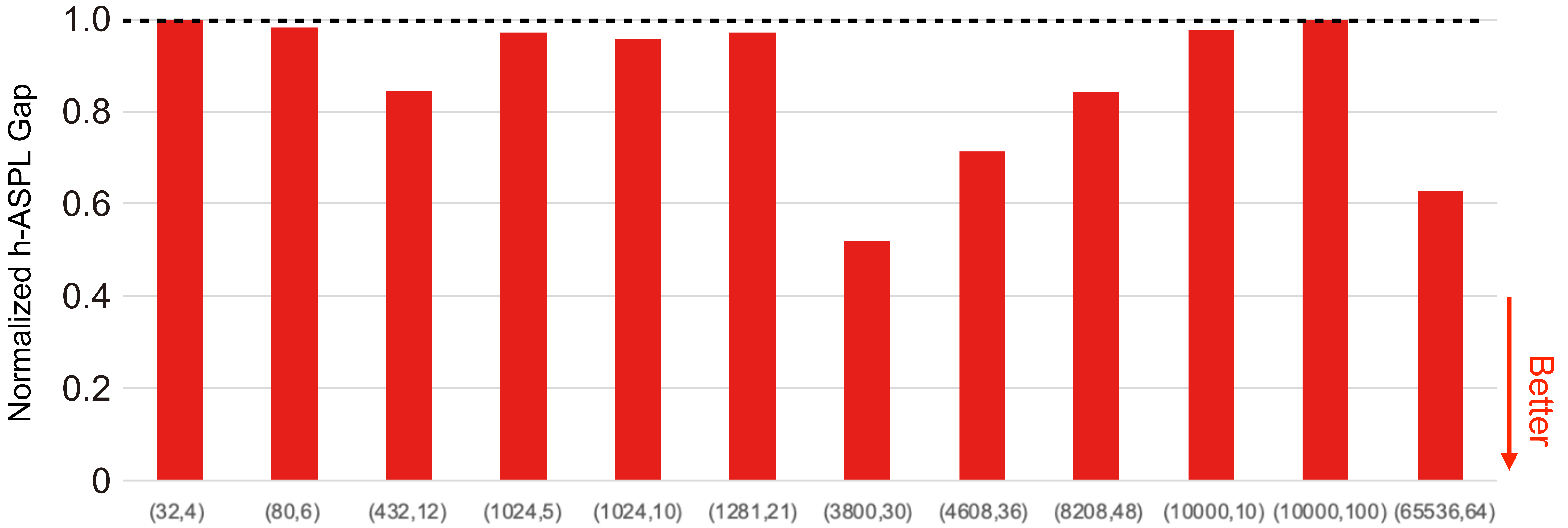
#hosts = 65536, radix = 64
#switches = 3072



- The h-ASPL Gap on the vertical axis is the difference between h-ASPL and the theoretical lower bound
- The larger the number of symmetries, the better the graph will be created

Example results (3/3)

- Normalize h-ASPL Gap with 1.0 when there is no symmetry
- **Results with symmetry are always better than those with no-symmetry**



Conclusion

- Proposal of optimization algorithm for ORP
- It was found that by giving symmetry, it is possible to generate a graph with a smaller h-ASPL